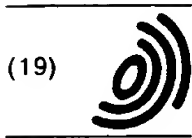


98/1082

B2



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 0 715 257 A1

(12) **DEMANDE DE BREVET EUROPEEN**

(43) Date de publication: 05.06.1996 Bulletin 1996/23
(51) Int Cl.⁶: G06F 9/46

(21) Numéro de dépôt: 95402573.0

(22) Date de dépôt: 17.11.1995

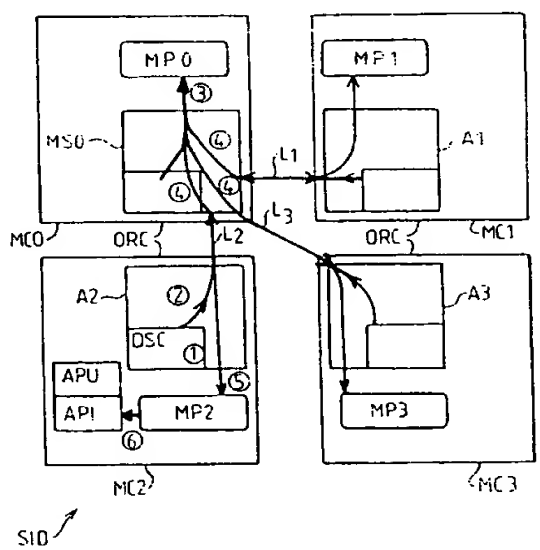
(84) Etats contractants désignés: DE ES FR GB IT SE	• Urbain, François F-75002 Paris (FR) • Saliba, Thérèse F-78190 Montigny Le Bretonneux (FR)
(30) Priorité: 30.11.1994 FR 9414386	
(71) Demandeur: BULL S.A. F-78430 Louveciennes (FR)	(74) Mandataire: Gouesmel, Daniel et al Direction de la Propriété Intellectuelle BULL SA, Poste courrier: LV59C18, 68 route de Versailles F-78430 Louveciennes (FR)
(72) Inventeurs: • Sitbon, Gérard F-94400 Vitry (FR)	

(54) **Outil d'aide à la répartition de la charge d'une application répartie**

(57) Outil (ORC) tournant sur les machines (MC0 à MC3) d'un système informatique (SID), destiné à répartir la charge sur chacune, caractérisé en ce qu'il comprend une pluralité de DAEMONS, l'un étant maître (MS0), les autres agents (A1 à A3),

- le maître (MS0) et les agents (A1 à A3) possédant chacun des moyens (MCC0 à MCC3) de calcul de la charge des machines sur lesquels ils tournent, et des moyens de mémorisation (MP0 à MP3) des données de charge du maître et des agents,
- le maître (MS0) contenant :
 - des moyens (MRC0 à MRC3) de collecte des données de charge de chacun des agents,
 - des moyens (MTC0) d'envoi des données de charge des autres agents à chacun de ceux-ci,
- chaque agent (A1 à A3) contenant :
 - des moyens (MRCC1 à MRCC3) de réception des données de charge des autres agents.

Applicable aux systèmes informatiques distribués.



EP 0 715 257 A1

Rest Available Copy

Description

La présente invention concerne un outil d'aide à la répartition de la charge d'une application répartie sur plusieurs machines appartenant à un système informatique distribué en réseau local.

La tendance actuelle du développement des systèmes informatiques est de former un tel système par association d'une pluralité de machines connectées entre elles par l'intermédiaire d'un réseau, de type local, par exemple. Tout utilisateur fait tourner des applications de types extrêmement variés sur cet ensemble de machines. Ces applications font appel à des services qui fournissent des informations nécessaires à la mise en oeuvre du ou des problèmes qu'elles traitent, lesquels sont offerts par tout ou partie de ces machines.

Lorsqu'une application en train de tourner requiert l'utilisation d'un service déterminé, dans la pratique courante, elle procède de la manière suivante :

- ou bien elle choisit de manière purement aléatoire la machine qui lui fournira ce service et lui confie ce travail,
- ou bien elle effectue un choix circulaire parmi toutes les machines, c'est-à-dire qu'elle confie tour à tour, toujours dans le même ordre temporel, le travail de fourniture des services qu'elle requiert successivement : ainsi, si le système possède trois machines, elle confiera à la machine No 1 le travail de fourniture des services qu'elle requiert en premier dans le temps, à la machine No 2, celui qu'elle requiert en second dans le temps, à la machine No 3, celui qu'elle requiert en troisième dans le temps et ainsi de suite dans l'ordre machine No 1, No 2, No 3, No 1, etc.

Que ce soit dans l'un ou l'autre des deux cas décrits ci-dessus, le travail d'aucune des machines n'est optimisé dans le temps, d'une part, et les possibilités de celles-ci en matière de débit et de performances ne sont utilisées que bien au-dessous de leur niveau maximum, d'autre part.

On connaît des solutions permettant de remédier à ces inconvénients : l'une de celles-ci est décrite dans la demande de brevet français No. 94 08764, déposée le 13/07/94, par la société demanderesse, sous le titre "système informatique ouvert à serveurs multiples". Dans un tel système formé par l'association d'un système central dit client et de plusieurs serveurs, chacun de ceux-ci calcule sa propre charge suivant des critères propres à chaque application tournant sur le client, ainsi que son évolution prévisible dans le temps et transmet ces deux facteurs au client. Ce dernier, lorsqu'une application déterminée requiert les services d'un serveur, choisit celui le moins chargé durant la période de temps où les services devront être rendus et lui confie le travail de fourniture des services demandés.

La présente invention constitue un perfectionnement, et une généralisation de la solution précédente.

Selon l'invention, l'outil au service d'une application répartie tournant sur les machines d'un système informatique distribué en réseau local, destiné à répartir la charge sur chacune de celles-ci, est caractérisé en ce qu'il comprend, une pluralité de modules informatiques, appelés DAEMONS, tournant sur ces dernières, dont l'un est appelé maître et les autres agents,

- le maître et les agents possédant chacun des moyens de calcul de la charge des machines sur lesquels ils tournent, à des premiers instants d'échantillonnage déterminés, et des moyens de mémorisation des données de charge du maître et des agents,
- le maître contenant :
 - des moyens de collecte des données de charge de chacun des agents, à des seconds instants d'échantillonnage déterminés,
 - des moyens d'envoi des données de charge des autres agents à chacun de ceux-ci,
- chaque agent contenant :
 - des moyens de réception des données de charge des autres agents,
- l'agent local le plus proche de l'application indiquant à celle-ci, sur requête de cette dernière, la machine la moins chargée, l'application prenant alors la décision de demander à cette machine d'exécuter les services dont elle a besoin.

D'autres caractéristiques et avantages de la présente invention apparaîtront dans la description suivante donnée à titre d'exemple non limitatif et en se référant aux dessins annexés. Sur ces dessins :

- La figure 1 montre un système informatique distribué incluant l'outil d'aide à la répartition de la charge selon l'invention,
- la figure 2 montre l'automate de répartition des rôles maître-agents entre les différents éléments constituant l'outil

d'aide à la répartition de la charge, selon l'invention.

1) CARACTERISTIQUES ESSENTIELLES DE L'OUTIL SELON L'INVENTION

A) structure :

Les différents éléments caractéristiques essentiels de l'outil ORC d'aide à la répartition de la charge (load balancing toolkit, en anglais) dans un système informatique distribué, selon l'invention -Pour simplifier, dans la suite du texte, il sera appelé "outil d'aide" - apparaissent à la figure 1.

Tel que montré à la figure 1, le système informatique distribué, de type quelconque, ici dénommé SID, comprend quatre machines informatiques de taille et de type quelconques, à savoir MC0, MC1, MC2, MC3. Chacune de ces machines -petits, moyens, gros ordinateurs- comprend les éléments habituels, à savoir un ou plusieurs processeurs centraux dits CPU (abréviation de Central Processor Unit, en anglais), des mémoires associées à ces derniers, des unités d'entrées/sorties (I/O units, en anglais), des moyens de connexion au réseau RE. Ce dernier est symboliquement représenté par des flèches à double sens représentant les liaisons de données entre les quatre machines MC0 à MC3, à la figure 1.

L'outil d'aide ORC proprement dit comprend le maître MS0, et les trois agents A1, A2, A3. Tout agent peut également être maître, selon des conditions qui seront explicitées par la suite. Aussi bien le maître que les agents sont constitués par des outils informatiques connus de l'homme du métier sous le nom de DAEMONS. Un DAEMON est un outil informatique ou entité tournant sur une machine, capable de répondre à une question.

A l'intérieur de chacune des machines MC0 à MC3, les DAEMONS MS0, A, A2, A3 sont associés respectivement à des mémoires partagées MP0, MP1, MP2, MP3. Chacune d'entre elles contient la charge de la machine correspondante mais également la charge des autres machines de SID.

On suppose, à la figure 1, que l'application répartie tourne sur la machine MC2 et qu'elle requiert donc des services fournis par les autres machines MC0, MC1, MC3. On désigne cette application par APU. L'endroit où se trouvent localisés le maître et les agents est indépendant de l'endroit où tourne APU.

B) fonctionnement :

Les grandes lignes de fonctionnement de l'outil ORC sont les suivantes, étant entendu que l'on suppose que, lors de l'établissement de la communication entre toutes les machines du système SID, il est établi que MS0 est le maître et que A1, A2, A3 sont les agents. On se réfère toujours à la figure 1, et notamment aux flèches et aux chiffres entourés qui les accompagnent, qui indiquent respectivement le sens du flux des informations qui circulent entre le maître et les agents, d'une part, et la séquence des opérations, d'autre part.

OPERATION 1 : Chaque agent ainsi que le maître recueillent, pour la machine sur laquelle ils tournent, à des intervalles de temps donnés qui constituent des premiers instants d'échantillonnages déterminés t_i , les données de charge de celle-ci, et ce pour chacun des éléments qui la constitue (charge de CPU, charge des mémoires associées, charge des entrées/sorties, charge du réseau, etc.). A partir de la charge de chaque élément, exprimée en pourcentage de la charge maximale admissible de celui-ci, on calcule la charge totale de la machine en question. Ceci est accompli par des moyens de calcul de charge, respectivement MCC0 pour MS0, MCC1 pour A1, MCC2 pour A2, MCC3 pour A3. Ces moyens sont tout simplement constitués par des programmes de calcul mettant en oeuvre le mode de calcul de la charge décrit plus bas, au paragraphe 2 : "Mode de calcul de la charge par chacun des agents". Ces moyens sont naturellement partie intégrante de chacun des maître et agents, MS0, A1 à A3 et, de ce fait, ne sont pas représentés en tant que tels à la figure 1, pour simplifier. Une fois la charge totale de la machine en question calculée, on obtient ainsi un ensemble de données statistiques sur la charge de la dite machine, à savoir DSC. A la figure 1, on n'a représenté cette opération que pour l'agent A2, pour des raisons évidentes de clarté de celle-ci.

OPERATION 2 : Les agents envoient à intervalles réguliers à MS0, les données statistiques de charge de la machine correspondante, par l'intermédiaire du réseau (pour A2, par l'intermédiaire de la liaison L2 entre MC2 et MC0).

OPERATION 3 : Le maître MS0 centralise, pratiquement à ces mêmes intervalles réguliers définis pour l'opération 2, qui constituent ainsi des seconds instants d'échantillonnages déterminés T_i , toutes les données statistiques de charge de tous les agents ainsi que les siennes propres au niveau de sa mémoire partagée associée, ici MP0. Cette centralisation est, de fait, une opération de collecte des données de charge. Elle est donc effectuée par des moyens de collecte des données de charge, respectivement MRC0 pour MS0, MRC1 pour A1, MRC2 pour A2, MRC3 pour A3, qui sont de fait des programmes de collecte intégrés dans le maître et dans chacun des agents A1 à A3 et ne sont donc pas représentés pour simplifier à la figure 1.

OPERATION 4 : Le maître MS0 envoie par l'intermédiaire de moyens d'envoi MTC0, toutes ces données à chaque agent A1, A2, A3 par l'intermédiaire du réseau RE, à savoir par l'intermédiaire des liaisons L1 entre MC0 et MC1, L2 entre MC0 et MC2, L3 entre MC0 et MC3. MTC0 est partie intégrante de MS0 et n'est donc pas représenté, pour

simplifier, à la figure 1.

OPERATION 5 : Chaque agent reçoit ces données de charge et les copie dans sa mémoire partagée associée. MP1 pour A1, MP2 pour A2, MP3 pour A3. Ceci est accompli par les moyens MRCC1 à MRCC3, respectivement pour A1 à A3, parties intégrantes de ceux-ci et non représentés pour simplifier à la figure 1.

OPERATION 6 : L'application APU va explorer la mémoire partagée de la machine sur laquelle elle tourne pour y rechercher la charge estimée pour chacune des machines, à l'instant où elle aura besoin qu'on lui rende des services déterminés, en déduit la machine la moins chargée à cet instant et demande à cette dernière de lui rendre ces services.

2) MODE DE CALCUL DE LA CHARGE PAR CHACUN DES AGENTS :

La description est faite sur un exemple de charges sur les éléments CPU, mémoire, Entrées/Sorties, et réseau RE.

La description du mode de calcul de la charge par chacun des moyens MCC0 à MCC3 est faite par référence aux tableaux 1 à 4 qui figurent en annexe 1 à la fin de la description et où les charges sont données en pourcentage.

Le calcul de la charge, pour chaque agent et maître, est identique à celui décrit dans la demande précitée. Il est rappelé brièvement ci-dessous.

La charge totale W_t d'un agent (et également du maître) est obtenue à partir de la formule suivante :

$$W_t = k_1 \cdot W_1 + k_2 \cdot W_2 + k_3 \cdot W_3 + k_4 \cdot W_4, \text{ où :}$$

- W_1 est le pourcentage d'utilisation dans le temps du processeur central de l'agent.
- W_2 est le pourcentage d'utilisation de la mémoire de l'agent, c'est-à-dire le rapport entre la capacité de mémoire réellement utilisée et la capacité totale de celle-ci.
- W_3 est le pourcentage d'utilisation du réseau par l'agent, c'est-à-dire le rapport entre le nombre d'informations émises et reçues par l'agent et le débit maximal admissible sur le réseau.
- W_4 est le pourcentage d'utilisation des unités d'entrée/sortie par l'agent.
- k_1, k_2, k_3, k_4 , sont des facteurs de pondération spécifiques du processeur, de la mémoire, du réseau, des entrées/sorties. Leur somme est égale à 1. Leurs valeurs dépendent de la nature de l'application en train de tourner, ici APU sur la machine MC2.

Les charges, W_1, W_2, W_3, W_4 sont mesurées et W_t calculée, ainsi qu'on peut le voir sur chacun des tableaux de l'annexe 1, à des instants d'échantillonnage déterminés $t_1, t_2, t_3, t_4, t_5, t_6, t_7$, etc., de période T (en fait les instants t_i , mentionnés plus haut lors de la description de l'opération 1).

Le tableau 1 donne un exemple de données de charge recueillies par un agent quelconque, par exemple A1, et relatives à la machine correspondante MC1, pour tous les instants t_1 à t_7 . Ces données sont bien entendu mémorisées dans la mémoire partagée MPI de la machine MC1 où tourne A1, avant qu'elles ne soient envoyées à MS0.

On peut donc voir sur ce tableau que, par exemple, W_1 est égal à 35 à l'instant t_1 , W_2 à 67 à l'instant t_4 , W_3 à 38 à t_6 , W_4 à 32 à t_7 et ainsi de suite.

Un programme de calcul API associé à APU, qui tourne sur MC2, applique ensuite, -pour les données de charge de chacun des agents et maître et qui, après exécution de l'opération 3, sont contenues dans la mémoire partagée MP0 de MC0 associée à MS0, -les facteurs de pondération k_1 à k_4 spécifiques des machines correspondantes pour l'application APU.

On obtient alors le tableau 2 qui montre, pour chacune des machines MC0 à MC4, la valeur des données de charge globale W_t , aux instants t_1 à t_7 . Ainsi, on peut voir que, pour MC0, W_t est égale à 56 à l'instant t_1 , 32 à t_2 , 67 à t_3 , etc.. Pour MC1, W_t est égale à 23 à t_1 , 34 à t_2 , etc. et ainsi de suite pour les autres machines.

L'étape suivante de calcul de la charge, pour toutes les machines, consiste à estimer, par extrapolation, par la méthode mathématique connue des moindres carrés, la valeur de la charge W_t estimée à l'instant $t_8 = (t_7 + T)$.

On obtient alors le tableau 3. On peut y lire, par exemple que la valeur estimée de la charge de MC0 à MC3, à cet instant t_8 est respectivement de 73, 82, 36, 76.

On applique ensuite à la charge totale de chaque machine, un coefficient de puissance C_p , spécifique de cette dernière, pour obtenir le taux de charge réel disponible C_1 de celle-ci, selon la formule :

$$C_1 = (100 - W_t (\text{estimée})) \cdot C_p.$$

En effet, il est important de tenir compte des caractéristiques de chaque machine, étant donné qu'on se trouve dans un milieu informatique hétérogène, où la puissance, la taille, et le type des machines qui le composent sont différents. Ainsi, si une machine est peu chargée, et si, en même temps sa puissance de traitement est insuffisante pour assurer les services que lui demande, à un moment donné, l'application APU, il est évident que c'est une autre machine qui doit assurer ces services. D'où la nécessité d'un facteur de correction pour définir la charge et, par suite, l'existence correspondante à cet effet du coefficient de puissance C_p .

Le coefficient C_p d'une machine donnée est calculé en faisant une synthèse de la puissance du processeur central CPU, de la capacité des mémoires, de la puissance de traitement des unités d'entrées/sorties, etc. Il est recalculé chaque fois que l'on modifie la configuration matérielle de la machine ou que l'on modifie son système d'exploitation (Operating System, en anglais). De même, chaque fois que la configuration générale du système informatique distribué SID est modifiée, tous les coefficients C_p de toutes les machines de ce dernier sont redéfinis. Un C_p égal à 1 correspond à une machine de type moyen, laquelle est définie par l'utilisateur.

On peut lire sur le tableau 4 des exemples de taux de charge réel disponible C_1 pour chaque machine MC0 à MC3. Ainsi, pour MC0, avec un taux de charge estimé de 73, un coefficient de puissance C_p de 2,5, on a un taux de charge réel disponible de 67,5. Les mêmes chiffres sont respectivement de 82, 2, 36 pour MC1 et ainsi de suite pour MC2 et MC3.

3) MODE D'ELECTION DU MAITRE MS0 :

La philosophie de base est que tout DAEMON tournant sur quelque machine que ce soit peut être maître. Il importe donc d'élaborer un mécanisme qui permette de définir lequel d'entre eux sera le maître et les conditions de son élection, d'une part, ainsi que les modalités de son remplacement s'il s'avère défaillant, d'autre part.

Le mécanisme d'élection doit s'assurer qu'au minimum 1 DAEMON est en train de tourner et que deux d'entre eux ne peuvent être simultanément maîtres (notamment s'ils démarrent au même instant).

Il se compose des 5 grandes phases suivantes :

Phase 1 : Lorsqu'un DAEMON démarre, il génère un identificateur unique ID conforme au protocole utilisé sur le réseau RE, par exemple conforme au protocole TCP-IP utilisé dans l'exemple de réalisation de l'invention décrit ici. Cet identificateur est composé de l'adresse Ethernet (Ethernet est la partie du protocole TCP-IP relative aux réseaux locaux qui est utilisée dans l'exemple de réalisation décrit ici, Ethernet étant bien entendu normalisé et donc connu de l'homme du métier), de l'instant d'émission de l'identificateur, et d'une valeur aléatoire. En même temps, il se place dans un état intermédiaire et envoie ces deux informations (état où il se trouve, ID) sur le réseau RE, à destination de toutes les machines de ce dernier.

Phase 2 : il attend de connaître les informations identiques provenant des autres DAEMONS, pendant un intervalle de temps déterminé T_r (de l'ordre de 5 à 10 secondes). Il est candidat à être maître.

Phase 3 : Dès qu'il les reçoit, il les analyse :

- Si elles proviennent d'un DAEMON qui, de fait, est maître, c'est-à-dire est considéré comme MS0, il se considère comme un agent.
- Si elles proviennent d'un DAEMON dans un état intermédiaire, alors il compare les identificateurs, le sien propre et celui qu'il reçoit :
 - si son propre identificateur est inférieur à celui qu'il reçoit, il conserve le droit d'être le maître MS0.
 - si son propre identificateur est supérieur ou égal à celui qu'il reçoit, il cède la place. Il réémet alors les deux dites informations (son propre ID, son état) et attend à nouveau des réponses, pendant le dit intervalle de temps T_r , encore appelé minuterie.

Phase 4 : Cet intervalle de temps étant écoulé, le DAEMON en question essaie à nouveau. Pour éviter la perte de messages, ce qui est toujours possible sur le réseau RE, on utilise la procédure suivante :

- L'émission et l'écoute des réponses sont répétées 5 fois,
- Si le DAEMON en question reçoit la réponse d'un autre DAEMON qui se révèle être un agent, il est sûr que le maître MS0 existe et il attend que la réponse de ce dernier lui parvienne.

Phase 5 : Lorsque les 5 répétitions ont été faites, et que le DAEMON en question n'a reçu aucune réponse de la part des autres DAEMONS, il décide alors qu'il est le maître MS0.

Quand l'un des trois agents A1 à A3 se rend compte que le maître MS0 ne communique plus avec lui, il entame la procédure ci-dessus dans toutes ses phases qui aboutit à l'élection d'un nouveau maître choisi parmi les trois.

De plus le maître notifie périodiquement son existence à toutes les machines du système SID. Si le maître détecte l'existence d'un autre maître, la procédure est reprise, par celui dont l'ID est le plus faible.

La figure 2 qui montre l'automate AUT de répartition des rôles maître-agents entre les différents DAEMONS tournant sur les machines de SID, permettra de mieux comprendre la succession des différentes phases 1 à 5 décrites ci-dessus.

Cet automate AUT comprend 5 états :

- état I0 : Le DAEMON en question émet les deux dites informations (son propre ID, son état), ce qui correspond à la phase 1.
- état I1 : Le dit DAEMON écoute les réponses des autres DAEMONS, ce qui correspond aux phases 2 et 3.
- état I2 : Le dit DAEMON est en attente de l'expiration du délai Tr, et d'une réponse éventuelle du maître MS0.
- état A : Le DAEMON en question devient un agent A1, A2, ou A3.
- état M : Le DAEMON en question devient le maître MS0.

Les événements qui correspondent à cet automate qui sont dénommés e1 à e8 sont les suivants :

- e1 : le DAEMON en question a diffusé son ID et son état et a fixé un délai Tr.
- e2 : Réception d'un ID, et l'identificateur ID local (celui du DAEMON en question) est inférieur à l'identificateur qu'il reçoit
- e3 : Réception d'un ID et l'ID local est supérieur ou égal à l'ID reçu.
- e4 : Le délai Tr a expiré.
- e5 : Le délai Tr a expiré et le nombre d'essais est inférieur à 5, ou bien un agent vient de répondre.
- e6 : le maître vient de répondre.
- e7 : le délai Tr a expiré et le nombre d'essais est égal à 5 et aucun agent n'a répondu.
- e8 : la connexion avec le maître est perdue.
- e9 : Détection par un maître de l'existence d'un autre maître d'ID supérieur.

ANNEXE 1

	t1	t2	t3	t4	t5	t6	t7
Charge CPU (W1)	35	12	42	73	92	65	33
Charge mémoire (W2)	45	32	33	67	46	32	40
Charge réseau (W3)	12	6	33	20	12	38	5
Charge entrée/sortie (W4)	25	30	56	46	78	44	32

Tableau 1: Exemple données de charge mémorisées dans toute mémoire partagée associée à un agent (exprimé en %)

	t1	t2	t3	t4	t5	t6	t7
MC0	56	32	67	63	79	82	54
MC1	23	34	45	56	67	62	79
MC2	32	38	34	42	35	32	36
MC3	96	94	79	82	74	79	68

Tableau 2: synthèse des séries de données de charge globale pour chaque machine

	t1	t2	t3	t4	t5	t6	t7	t8 = t7 + T
MC0	56	32	67	63	79	82	54	estimée 73
MC1	23	34	45	56	67	62	79	estimée 82
MC2	32	38	34	42	35	32	36	estimée 36
MC3	96	94	79	82	74	79	68	estimée 73

Tableau 3: extrapolation de la valeur de la charge globale après un délai de T pour chaque machine

	MC0	MC1	MC2	MC3
Taux de charge estimé	73	82	36	76
Coefficient de puissance	2.5	2	0.8	1.5
Coefficient de la charge disponible (100 - charge estimée) * coefficient de puissance	67.5	36	51.2	36

Tableau 4: application du coefficient de puissance, comparaison et sélection

Revendications

- 5 1. Outil (ORC) au service d'une application (APU) répartie tournant sur les machines (MC0 à MC3) d'un système informatique (SID) distribué en réseau local (RE), destiné à répartir la charge sur chacune de celles-ci, caractérisé en ce qu'il comprend une pluralité de modules informatiques (MS0, A1 à A3), appelés DAEMONS, tournant sur ces dernières, dont l'un est appelé maître (MS0) et les autres agents (A1 à A3),
 - 10 - le maître (MS0) et les agents (A1 à A3) possédant chacun des moyens (MCC0 à MCC3) de calcul de la charge des machines sur lesquels ils tournent, à des premiers instants d'échantillonnage déterminés t_i , et des moyens de mémorisation (MP0 à MP3) des données de charge du maître et des agents,
 - le maître (MS0) contenant :
 - 15 - des moyens (MRC0 à MRC3) de collecte des données de charge de chacun des agents, à des seconds instants d'échantillonnages déterminés T_i ,
 - des moyens (MTC0) d'envoi des données de charge des autres agents à chacun de ceux-ci,
 - chaque agent (A1 à A3) contenant :
 - 20 - des moyens (MRCC1 à MRCC3) de réception des données de charge des autres agents,
 - l'agent local le plus proche de l'application indiquant à celle-ci, sur requête de cette dernière, la machine la moins chargée, l'application prenant alors la décision de demander à cette machine d'exécuter les services dont elle a besoin.
- 25 2. Outil selon la revendication 1, caractérisé en ce qu'il comprend des moyens d'élection (AUT) d'un maître parmi les DAEMONS, assurant l'existence et l'unicité de ce maître, au démarrage de ces derniers et suite à la perte d'un maître en cours d'exécution de l'outil (ORC).
- 30 3. Outil selon la revendication 1, caractérisé en ce qu'il comprend des moyens (AUT, e8) permettant d'assurer la continuité de service rendu par l'outil à l'application, suite à une panne affectant au moins une machine du système informatique.
- 35 4. Outil selon la revendication 1, caractérisé en ce qu'il comprend des moyens (AUT, e1, état 10) de découverte automatique du réseau des machines lui permettant de récupérer les adresses de chacune des machines, au démarrage des DAEMONS.
5. Procédé de mise en oeuvre de l'outil selon la revendication 1, caractérisé en ce qu'il comprend les opérations 1 à 6 suivantes :
 - 40 1) Les agents et le maître (MS0, A1 à A3) recueillent, pour la machine sur laquelle ils tournent, les données de charge de celles-ci, aux premiers intervalles de temps t_i , et les moyens de calcul de charge (MCC0 à MCC3) calculent la charge totale de celle-ci à partir des dites données et de sa puissance.
 - 2) Les agents (A1 à A3) envoient au maître (MS0), aux seconds intervalles de temps T_i , les données de charge de la machine correspondante.
 - 45 3) Le maître (MS0), à ces mêmes seconds instants T_i , centralise les données de charge de tous les agents et les siennes propres, par l'intermédiaire des dits moyens de collecte (MRC0 à MRC3). 4) Le maître (MS0) envoie, par l'intermédiaire des moyens d'envois (MTC0), toutes ces données à chaque agent (A1 à A3).
 - 5) Chaque agent (A1 à A3) copie toutes ces données de charge dans sa mémoire partagée associée (MP1 à MP3).
 - 50 6) L'application (APU) recherche, pour l'instant où elle estime avoir besoin qu'on lui rende un service déterminé, et dans la mémoire partagée (MP0 à MP3) de la machine où elle tourne, la charge estimée de chacune des machines (MC0 à MC3), en déduit celle la moins chargée à ce même instant et lui demande alors de lui rendre le dit service.

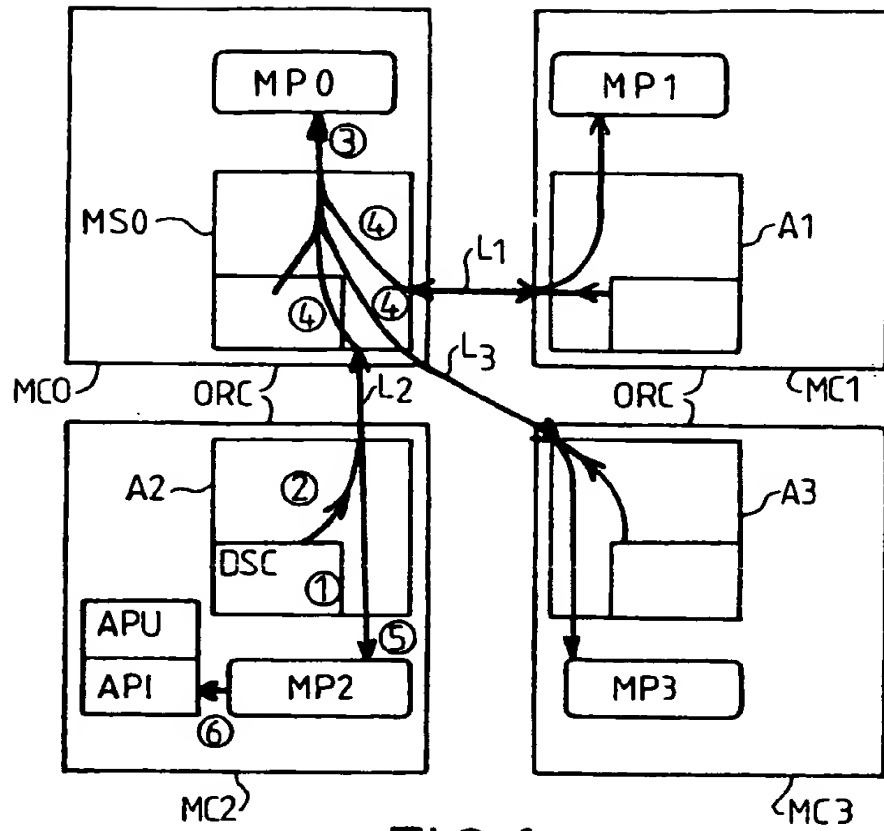


FIG. 1

SID ↗

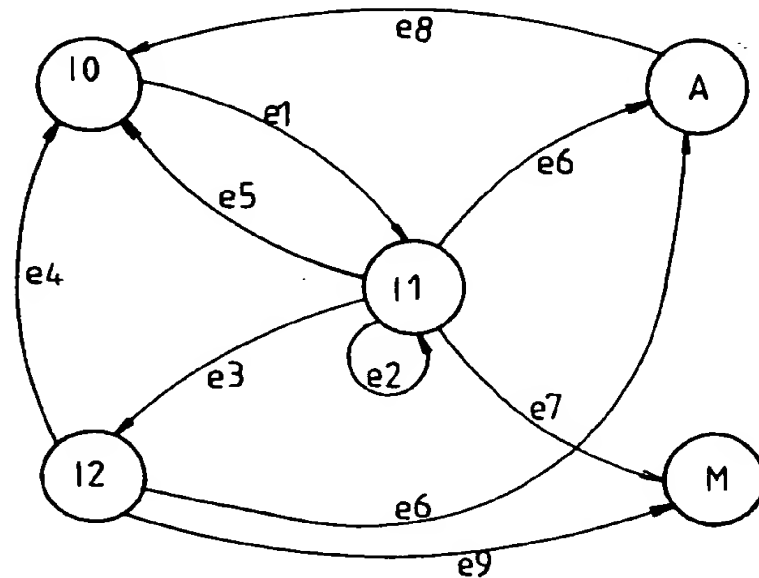


FIG. 2

This Page Blank (uspto)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)